

# Python : développement réseau avec TCP, HTTP et APIs

## Développement ▶ Python

### Présentation

Python est fréquemment utilisé pour des applications liées à des technologies disposant de peu de ressources, comme des systèmes embarqués, des équipements liés à l'internet des objets (IoT), des capteurs avec une connexion réseau. Pour interagir avec ces équipements, que ce soit pour envoyer ou recevoir des informations, il est nécessaire de disposer de connaissances en développement réseau. Différentes problématiques émergent alors : comment échanger de l'information efficacement, comment se prémunir de dysfonctionnements éventuels, comment recevoir un flux d'informations. Savoir répondre à ces questions améliore de manière générale la qualité et l'empreinte des applications.

Au cours de cette formation, vous apprendrez tout ce dont vous avez besoin pour développer des programmes interagissant sur le réseau, que ce soit avec les protocoles TCP, UDP ou HTTP. Vous découvrirez comment échanger et recevoir de l'information, déclencher des actions à distance et gérer les aléas du réseau. Enfin, vous découvrirez des modèles de programmation brouillant les définitions de client et de serveur, pour vous initier au développement de microservices.

### Objectifs

- ▶ Savoir envoyer et recevoir de l'information via TCP, UDP et HTTP
- ▶ Savoir créer des applications client et serveur avec Python
- ▶ Savoir utiliser les modules socket, requests et le microframework Flask
- ▶ Savoir sérialiser les données pour la communication inter-programmes
- ▶ Apprendre à être résilient vis-à-vis des problématiques réseau : volume, aléas, flux
- ▶ Découvrir le développement d'applications et d'APIs distribuées
- ▶ Savoir développer un microservice Python

### Moyens pédagogiques

Formation alternant la théorie et la pratique, avec un intervenant présentant la formation en direct. Possibilité de réaliser cette formation en face-à-face dans nos locaux, dans vos locaux d'entreprise ou à distance via visioconférence.

Une fois terminée, vous disposerez d'une attestation de suivi de la formation, ainsi que des contenus de cours relatifs au contenu présenté.

### Evaluation des acquis

Questionnaires à choix multiples (QCM) en fin de formation.

L'intervenant formulera également une appréciation sur vos acquis pour l'entièreté de la formation.

**Durée** : 28 heures (4 jours)

**Référence** : DE-118

**Niveau** : Intermédiaire

**Formation inter-entreprises** :

Formation en présentiel et distanciel

**Tarif** : 2400 € HT

**Formation intra-entreprises** :

Formation disponible et adaptable sur demande

**Tarif** : sur devis

**Contact**

**Anthony DUPUIS**

Conseiller formation

07 68 53 96 11

anthony@ls-a.fr

**Public** :

Toute personne souhaitant développer des applications logicielles résilientes ou améliorer ses compétences en programmation Python.

**Avez-vous les prérequis ?**

Une connaissance équivalente à la formation DE-090 (Python : les fondamentaux) est nécessaire

**Dernière modification** :

08/11/2024

# Python : développement réseau avec TCP, HTTP et APIs

Développement ▶ Python

## Programme de la formation

### 1- Notions de réseau

Types de réseau et interfaces réseau  
Les principaux protocoles  
Adressage IP, DHCP, DNS

### 2- Utiliser TCP et UDP avec Python

Fonctionnement TCP et UDP : les différences  
Présentation des « sockets » et de la bibliothèque socket  
« Hello World » avec les sockets Python  
Modèle client-serveur  
Problématiques liées au contenu transféré et méthodologies de lecture

### 3- Echanger des messages complexes entre client et serveur

Sérialisation de contenu avec la bibliothèque json  
Créer des modèles de messages entre le client et le serveur  
Créer un serveur attendant des actions  
Notions liées à RPC (Remote Procedure Call)

### 4- Echanger des messages avec HTTP

Présentation des composantes d'HTTP : méthodes, en-têtes, contenu  
Envoyer des requêtes HTTP avec le module requests  
Recevoir des requêtes HTTP avec Flask  
Traiter les erreurs avec les codes de statut

### 5- Sortir du modèle client-serveur

Créer un service recevant et envoyant des requêtes (HTTP ou TCP)  
Présentation du concept de microservice : avantages et inconvénients  
Créer des microservices TCP et HTTP  
Notions d'indisponibilité du service distant, tolérance aux pannes  
Notions de transfert de requêtes, de proxy et d'API Gateway